VIPER

VOLUME 2, ISSUE 4 AN ARESCO PUBLICATION OCTOBER 1979, $2.00

TABLE OF CONTENTS

Please note that there is an order form on page 2.04.28 for PIPS volumes 1 - 3. You don't have to use this form to order; it was included so you can see what information we need to process your orders. You can use the order form on page 2.04 .29 - but do use the PIPS form for an information guide.

SUBSCRIPTION RATES, ADVERTISING RATES, & OTHER INFORMATION

THE VIPER IS PUBLISHED TEN TIMES PER YEAR BY ARESCO INC., AT 6303 GOLDEN HOOK, COLUMBIA MD 21044. MAILED TO SUBSCRIBERS ON THE 15TH DAY OF EACH MONTH EXCEPT JUNE AND DECEMBER. SINGLE COPY PRICE IS $2 AND SUBSCRIPTION PRICE IS $15 FOR ALL 10 ISSUES OF THE CURRENT VOLUME. SUBSCRIPTIONS DO NOT CARRY OVER FROM ONE VOLUME TO THE NEXT. SUBSCRIBERS WHO WISH TO ORDER LESS THAN THE FULL VOLUME SHOULD SEND $2 FOR EACH ISSUE DESIRED. RENEWALS ARE ACCEPTED DURING THE LAST TWO MONTHS OF THE CURRENT VOLUME YEAR, AND THE FIRST ISSUE OF EACH VOLUME IS PUBLISHED IN JULY.
APPLICATION TO MAIL AT SECOND CLASS POSTAGE RATES IS PENDING IN COLUMBIA MD 21045. POSTMASTER: SEND ALL ADDRESS CHANGES TO THE VIPER, BOX 1142, COLUMBIA MD 21044.
THE VIPER IS AN ARESCO INC. PUBLICATION. EDITOR: TERRY LAUDEREAU, CO-EDITOR: RICK SIMPSON. CORRESPONDING EDITOR IS TOM SWAN.
READERS ARE ENCOURAGED TO SUBMIT ARTICLES OF GENERAL INTEREST TO VIP OWNERS FOR PUBLICATION IN THE VIPER. MATERIAL SUBMITTED WILL BE CONSIDERED FREE OF COPYRIGHT RESTRICTIONS. THOSE SUBMISSIONS RECEIVED BY THE 1ST DAY OF A MONTH WILL BE CONSIDERED FOR INCLUSION IN THAT MONTH'S ISSUE.

SUBSCRIPTION RATES:
USA RESIDENTS: $15/10 ISSUES. NON USA RESIDENTS SHOULD INCLUDE AN ADDITIONAL $10 FOR AIR MAIL POSTAGE. IF SUCH POSTAGE IS NOT INCLUDED WITH THE ORDER, THE NEWSLETTER WILL BE SENT SECOND CLASS WITH THE REMAINDER OF THE MAILING. PURCHASE ORDERS WILL NOT BE ACCEPTED. IF SUBSCRIBER SPECIFIES UPS COD THE FIRST ISSUE WILL BE SENT COD WITH THE COLLECTIBLE AMOUNT TO BE $15 PLUS A $1 COD CHARGE PLUS SHIPPING COSTS. PERSONAL CHECKS, MASTER CHARGE, AND VISA ORDERS ARE ACCEPTED. CHECKS DRAWN ON FOREIGN BANKS SHOULD BE PAYABLE IN U.S. DOLLARS.

ADVERTISING RATES:
CLASSIFIED ADVERTISING: $10/3 LINES OF 60 CHARACTERS EACH.

PAGE RATES:

FULL PAGE	$85
HALF PAGE	$45
QUARTER PAGE	$25

PAYMENT MUST ACCOMPANY ADS. ADS SHOULD BE CAMERA READY POSITIVE ARTWORK. IF OUR PRINTER CHARGES US EXTRA FOR PREPARING YOUR AD COPY, WE WILL BILL YOU AT COST + 10%.

DEALERS ARE INVITED TO REQUEST THE ARESCO INC. DEALER INFORMATION PACKAGE BY WRITING IN CARE OF DEALER SERVICES.

Rick- The VIP is a delightfully simple machine to operate. I use mine as a peripheral to a Heathkit H II. By connecting the VIP I/O slot to a Parallel Interface Module in the H II, I can trasfer programs and data from one to the other. Thus I can store H II programs on the VIP cassette which is faster and easier than using paper tape.

I also needed a DMA controller for a "Reticon" and again the VIP was just the thing. The Reticon is a solid state optical scanner which puts out video data through an ADC. I made a controller using 5 CMOS chips which transfers the data to VIP memory and thence through the P.I.M. to the H II for munber crunching. The data comes off the ADC a bit slower than the DMA machine cycles of the VIP so I strech the TPB pulse using the WAIT line to synchronize the two.

I would be glad to send more details if anyone is interested. -Bill Lonley

Terry- On page 2.02.02 you mention pressing 'E' key to get to machine language programming with 'BASIC' board inserted. This is fine, you can write, read, tape, and load but you have no facility to run program. After a call to RCA, I received over the phone a modification (connecting a SW) to enable the original keyboard without always removing 'BASIC' board. Unfortunately, I am not at home and cannot send modifications. If you are interested let me know and I will send it in October when I return home. -Hank Ruhl, Jr.

Terry- Since RCA informed me that my keyboard will not be delivered until late October/early November, I have ordered the parts to convert an old Univac keypunch board to ASCII. Until that time, I won't know for sure, but as it stands now - no chip 8 program will fun while the tine BASIC is plugged in. Pressing "E" will give you access to the monitor, but if you throw the reset to start your program, you're back in BASIC. Pressing "3" and the reset gains a blank screen. An "E" on the aux keypad give a space in BASIC.

After the keyboard, I'll finish a half-built joystick mod (ah, Space Wars!), then perhaps a TV typewritter.

So much hardware - to think I bought the VIP because I like doing software. -Louis Schlekau III

Rick - After reading the August VIPER, I have a few comments:

For C Thompson - there are articles for using the ELF as an automatic keyer and message sender for code in the January and February 1979 issues of "73" magazine. If he can't get the magazines, I'll send him copies of the articles. They don't tell him how to get from CW to a display, but they're a place to start.

For John B Sewell - I've found almost any tape recorder and tape cassette works. Don't put the recorder close to a monitor, though. Also, when I plugged in the color board, I had to move the tape recorder at least 2 feet away from the VIP. Demagnetize the heads often. All this may not work for you - but it sure helped me. I've had really good results with a Radio Shack TV RF interface. (After I got the color board, I needed something for color. Usually I use a modified TV when using black and white.

For anyone: Any articles on 1802 systems are fine with me. I may buy some ELF boards for my VIP if the ELF suppliers stay ahead of RCA in offering keyboards, BASIC, or orther improvements.

I also bought a Cuddly Software program. It was worth the money. The back-up service is great if you have any problems. - Jim Brooks, 3107 10th NW, Albuquerque NM 87101

For those of you who have discovered more than our fair share of typos: We would not have to retype if you could send your articles (especially code) typed in the format you see in these pages. Hand printed code is reproducible, too, if it's clear and neat enough to read. Printer output is often not in dark ink, and we all know what happens when we try to reproduce that (Remember Barrett's article in V1 #4?). Also - if we can get one of the Trendcom printers ourselves, we'll be able to do hex dumps - but that's not commented code. ALWAYS feel welcome to call us if you have a problem reading, entering, or using material we use here! - Terry

Rick- I just finished building a joystick and interface for the VIP that VIPER readers might be interested in. It takes five minutes to build and costs less than forty-nine cents!

The key to its simplicity and low cost is the fact that it uses a mechanical interface rather than an electrical interface. It is nothing more than a flat disk with four small "feet" on the bottom and a stick on the top. The "feet" are places so that they sit on the 2, 4, 6, and 8 keys of the VIP keyboard. A slight pressure on the stick in any of the four directions causes a "foot" to depress the appropriate key. Because the VIP keypad has a light touch, the joystick is quite sensitive.

For the reet I used self-adhesive sork disks that are sold in hardware stores. They are normally used on the bases of ash trays, vases, clock radios, etc. to avoid marring table top finishes. I used 3/8 inch disks, and used two for each foot (to make each foot a bit deeper, which gives the joystick setter action.) The package of 24 feet cost me forty-nine cents.

For the flat disk and stick I used an empty spool that had held 50 feet of 24 guage speaker wire. This eliminated the gluing operation to attach the stick to the disk.

I have been using this joystick with Tom Swan's Space War, and it has really improved the entertainment value of the game.

I've already thought of an improvement to the joystick to make the space simulation more interesting. If an empty spool of sewing thread is glued to a flat disk (try a one quart orange juice jar cover) and a hole is drilled through the disk to match the spool hole, then a small stick can pass through the center of the joystick and provide plunger action on key 5. If the program is changed to make key 5 the fire button, then the joystick can provide both navigation and fire control.

I hope that other VIPER readers will try this idea. For forty-nine cents, you can't find a better peripheral. -Bob Kantor

ATTENTION PLEASE!

The article in this issue (A cheap printer for the VIP) by Ruben de la Pena was accompanied by a schematic, but it was apparently a copy of a copy. We couldn't get it to reproduce well enough to tell what it was, so we left it out altogether. If you'd like to see one, write to Ruben and ask him if he'll send it to you. We don't know that he will - our letter hasn't been answered yet (on 9/28, as we go to press with this issue). The bottom line is that you can get a printer kit for less than $120 from him; and that sounds like it might be worth looking into!

KEYBOARD KONTROL

by

Tom Swan

Taxco, Gro., MEXICO

In the September '78 issue of VIPER, Rick Simpson showed us how to add input/output instructions to the Chip-8 interpreter by replacing the BMMM instruction with a new machine language subroutine. RCA's new keyboard is compatible with these modifications and you may input ASCII codes for keys pressed from the I/O port with a single B1X1 instruction where the "X" indicates the Chip-8 variable you want to hold the input. Using the B1X1 instruction will cause your program to halt until you press and release a key on the keyboard. VX will then equal the ASCII code of the key that was pressed.

But I was interested in using my keyboard -- RCA's prototype model that Rick was kind enough to send me so I could put some software together for fellow Vipers as soon as possible -- in two-player action-type games. The B1X1 instruction just wouldn't do. Even though its use of the EF-4 flag meant there was no conflict with the VIP hexpad, the instruction could not be used, for instance, to move two tanks around the screen with one player working the hexpad and the other on the keyboard.

I didn't want to change Rick's modifications, so I found a way to add yet one more instruction to Chip-8 that allows an ASCII keyboard, such as the RCA model, to be used in two player games.

First you need to make Rick's modifications to a copy of the normal Chip-8 Interpreter. The correct listing for these changes appears in the February '79 issue of VIPER on the inside back cover. This modification is named Chip-8 I. (If you do not have a copy of that issue, these new modifications here will work as described -- but you won't have Rick's Chip-8 I instructions to use.)

Next, make the changes listed below in order to upgrade Chip-8 I to Chip-8 II. These changes in no way affect the Chip-8 instruction set (of course the BMMM is still out of action), and the I/O instructions of Chip-8 I are used in the same way as before.

Chip-8 II, however, gives you another input instruction

which operates in much the same way as the B1X1. The new instruction will cause the byte at the input port to be placed in VX provided a key is being pressed when the instruction is encountered. Here is the new instruction:

FX00 - Input byte @ port into VX

The difference from the B1X1 is that your program will not halt at the point the FX00 is used. If a keyboard key is being pressed, its ASCII code will be placed in VX. If no key is being pressed, VX will not be changed, and the program will continue on to the next instruction in sequence.

To use the FX00 for two-player games, first initialize any variable to equal "00," execute FX00, then test VX to see if a key had been pressed. For example, you could program the following two instructions:

```
6000 V0=00 (Initialize V0)
F000 V0=Keyboard key ASCII code if pressed
```

Next, test V0 to see if a particular key had been pressed. If you wanted a certain action -- maybe a pong paddle to move -- when the keyboard Key A was pressed, you would then test V0 to see if it equalled the ASCII code for the letter A. After the above two instructions, this would do the job:

```
4041 Skip if V0≠41, ASCII for "A"
2MMM Do sub - Move Paddle
```

Of course, you don't have to use a subroutine call, I only show it for the example. The initializing of the variable, by the way, should not be left out. If it were and the program looped back to this same point, the same action would again be called even if Key A were not being pressed again.

HOW SHE WORKS

To my surprise, I found a way to shorten the Chip-8 Interpreter by two bytes making enough room at 0100-0105 to include the new FX00 instruction. Two bytes may not seem like very much, but with an interpreter packed as tightly as Chip-8, finding two new free addresses is as rare as winter rain here in southern Mexico. (We just broke a 6-month dry spell with a wonderful downpour -- but that's normal for here, though to be sure, I toss a peso into the pond at the foot of Tlaloc the Rain god in Mexico City every once in a while. Can't hurt.)

The change at 006F accomplishes this little extra squeeze by redirecting the decoding of FXNN type instructions from its former position at 0105 to a new spot at 019C. The FXNN's work by having NN set equal to the low portion of the address (on the 0100-01FF page) where the needed subroutine is stored. This subroutine is the one that performs a function for whatever FXNN instruction has been programmed. For instance, the subroutine to set the tone duration equal to VX is located at 0118 to 011A. The 18 in the second half of the address corresponds with the Chip-8 instruction FX18, set tone duration. This is no coincidence -- the 18 in the instruction tells the interpreter which subroutine to call on that memory page.

Notice that the two instructions (before you make the changes below) at 0105 and 0106 are the same as the two instructions at 019C and 019D. What I have done is to "steal" the two at 019C to decode the FXNN instructions, opening the two at 0105 and 0106 for the new FX00 which then just fits. At 019C, the second half of the FXNN Chip-8 instruction is placed in R3.0, the program counter, which causes a jump to the desired subroutine. Be careful in the future if you further modify the interpreter not to change the instructions at 019C and 019D. If you do, all of the FXNN instructions would be disabled!

The new subroutine at 0100-0105 is very simple. If a key is being pressed, X is set equal to 6 so that R(X) points to the Chip-8 variable addressed by R(6). When the input instruction, 6B, is executed to bring in the byte from the input port generated by the keyboard, that byte is automatically gated to the memory byte pointed to by R(X). In this case R6 addresses the storage byte for VX specified in the FX00 instruction, and the input is stored as the new VX value. This only requires 6 bytes to accomplish further demonstrating how the 1802 microprocessor allows for tight interpreter design.

The sample program below is not really much of a game, but then I only intend it as a starting point for you. Keys 2; 4; 6; and 8 on the hexpad move one square around on the screen while Keys W; A; S; and Z on the keyboard move the other square. That's all -- and I don't expect any applause. How about changing the two squares to tanks? Or rocket ships? Insert scoring, destruction sequences, beepers, etc., and you've got a full scale two-player arcade game to add to your library, plus a way to demonstrate your new keyboard to your friends.

If you care to write, I'd appreciate your comments, criticisms, questions -- or just to talk. Please enclose an SASE (self-addressed stamped envelope -- U.S.A. postage is OK) so I can get back to you as soon as possible. (It's no more expensive to here -- a standard 15¢ stamp will do just fine by the way.) Best of luck with your programming!

THE MODIFICATIONS

Make the following changes to a copy of Chip-8 I, and call the new version Chip-8 II (in case we refer to it at a later date).

```
0100  37  B4        ;Branch if EF-4=1 - (Key is pressed)
  01  03            ;to 0103
  02  D4  SEP   R4 ;Else return -- no key is pressed
  03  E6  SEX   6  ;Set X=6, R(X) (R6) points to VX
  04  6B  INP       ;Input byte from port.  Place in VX
  05  D4  SEP   R4 ;Return to 0042 for next instruction
```

Also make the following change.

```
006F  9C            ;New address of FXNN decoding @ 019C
```

This completes the modifications for Chip-8 II. The following program demonstrates the use of the new FX00 instruction.

TWO-PLAYER KEYBOARD TEST PROGRAM

```
0200  BEGIN :6A00 ;VA=00 -- Preset VX coordinate block #1
  02        6B00 ;VB=00 -- Preset VY coordinate block #1
  04        6C00 ;VC=00 -- Preset VX coordinate block #2
  06        6D10 ;VD=10 -- Preset VY coordinate block #2
  08        A25A ;BLOCK -- Set "I" to display pattern @ 025A
  0A        DAB5 ;SHOW1 -- Show block #1 @ VA VB
  0C        DCD5 ;SHOW2 -- Show block #2 @ VC VD
  0E  LOOP  :6000 ;V0=00 -- Let V0=00 - VX direction adder

0210        6100 ;V1=00 -- Let V1=00 - VY direction adder
  12        6202 ;V2=02 -- Let V2=02 - For hexpad key test
  14  KEY   :E2A1 ;SK≠KY -- Skip if hex key pressed ≠ V2 value
  16        1220  MOVE  -- Else go to 0220 to move block #1
  18        7202 ;V2+02 -- Add 02 to V2 for next hex key test
  1A        320A ;SK=0A -- But if V2=0A, skip to continue on
  1C        1214  KEY   -- If V2≠0A yet, loop back to 0214
  1E        6200 ;V2=00 -- Set V2=00 only if no key was pressed

0220  MOVE  :4202 ;SK≠02 -- Skip if not Key 2
  22        61FF ;V1=FF -- V1=FF if Key 2 -- move up
  24        4204 ;SK≠04 -- Skip if not Key 4
  26        60FF ;V0=FF -- V0=FF if Key 4 -- move left
  28        4206 ;SK≠06 -- Skip if not Key 6
  2A        6001 ;V0=01 -- V0=01 if Key 6 -- move right
```

```
022C          4208 ;SK≠08 -- Skip if not Key 8
  2E          6101 ;V1=01 -- V1=01 if Key 8 -- move down

0230          DAB5 ;ERASE -- Erase block #1 @ old VC VD
  32          8A04 ;VA+V0 -- Add V0 to VA for new VX
  34          8B14 ;VB+V1 -- Add V1 to VB for new VY
  36          DAB5 ;SHOW  -- Display block #1 @ new VA VB
  38          6000 ;V0=00 -- Let V0=00 -- VX direction adder
  3A          6100 ;V1=00 -- Let V1=00 -- VY direction adder
  3C          6200 ;V2=00 -- Let V2=00 -- For keyboard test
  3E          F200 ;INPUT -- V2=ASCII input if key is pressed

0240          4257 ;SK≠57 -- Skip if not Key W (ASCII 57)
  42          61FF ;V1=FF -- V1=FF if Key W -- move up
  44          4241 ;SK≠41 -- Skip if not Key A (ASCII 41)
  46          60FF ;V0=FF -- V0=FF if Key A -- move left
  48          4253 ;SK≠53 -- Skip if not Key S (ASCII 53)
  4A          6001 ;V0=01 -- V0=01 if Key S -- move right
  4C          425A ;SK≠5A -- Skip if not Key Z (ASCII 5A)
  4E          6101 ;V1=01 -- V1=01 if Key Z -- move down

0250          DCD5 ;ERASE -- Erase block #2 @ old VC VD
  52          8C04 ;VC+V0 -- Add V0 to VC for new VX
  54          8D14 ;VD+V1 -- Add V1 to VD for new VY
  56          DCD5 ;SHOW  -- Display block #2 @ new VC VD
  58          120E  LOOP  -- Go to 020E to loop
  5A  BLOCK  ;F0F0 ;BITS  -- Bit pattern for block #1 & #2
  5C          F0F0        --       "             "
  5E          F0F0        --       "             "
```

ANSWER TO LAST MONTH'S LITTLE LOOPS

Last month I suggested writing a subroutine that would use the instruction, BMMM ;GO TO OMMM + the value in V0, to set the "I" pointer to a certain bit pattern depending on the value in V0. If you wrote the routine, you may have noticed that the same thing may be accomplished in other ways -- no doubt you have done this in your game programs.

Supposing you program the following subroutine:

```
BMMM SUB

0400  SET I ;B400 ;GO+V0 -- Go to 0400 plus value in V0
  02          AXXX  BITS1 -- I = bit pattern #1 if V0=02
  04          00EE ;RET   -- Return
  06          AXXX  BITS2 -- I = bit pattern #2 if V0=06
  08          00EE ;RET   -- Return
```

Calling this subroutine will cause the I pointer to address the bit pattern #1 if V0=02 or the bit pattern #2 if V0=06. If V0=08, then only the return instruction at 0408 is executed, and "I" does not change.

The usual way to program such a routine, however, is as follows:

```
SKIP SUB

0400  SET I :4002 ;SK≠02 -- Skip if V0≠02
  02         AXXX  BITS1 -- I= bit pattern #1 if V0=02
  04         4006 ;SK≠06 -- Skip if V0≠06
  06         AXXX  BITS2 -- I= bit pattern #2 if V0=06
  08         00EE ;RET   -- Return
```

The two subroutines are the same apparent length and seemingly produce the identical results. If V0 is set equal to a certain key press for example, then a call to either sub would set "I" to bit pattern #1 if Key 2 was pressed or bit pattern #2 if it had been Key 6.

The danger with the first routine, the BMMM Sub, is that if any keys <u>other than</u> 0, 2, 4, 6, or 8 are pressed, a possible crash would result as control would pass beyond the end of the subroutine with a value higher than 8. In addition if V0 happened to be an odd number, the branch could end up in the middle of a Chip-8 instruction somewhere -- and I don't have to guess at the results of that condition. The second sub will handle any value of V0 without this danger.

Then why even suggest the first subroutine? The reason is at first not obvious, but the BMMM sub has two advantages including one hidden operation not included in the second sub.

First, the BMMM sub is one instruction shorter than the second! Now I can count to 5 as well as the next programmer (usually), and obviously each subroutine contains 5 lines of Chip-8 instructions. But, the BMMM subroutine will <u>execute</u> one less instruction than the second in order to produce the same result! Pick some values (even-numbered ones) for V0 and follow the program flow through each subroutine counting the number of instructions that will be <u>executed</u> for each value. You will find that the BMMM subroutine will always execute three instructions to set "I" while the second subroutine needs four instructions to do the same job. The minimum for the BMMM subroutine is two executions while the second subroutine takes at least three to return. In fact, no matter how many different bit patterns are involved or how long the sub, the BMMM sub will never require more than three instructions to execute! But as the length of the second subroutine grows,

so does the time required for it to operate. In action games, this could be an important difference.

If you are interested, the formula for the number of Chip-8 instructions that will be executed by using the approach in the second sub -- the Skip Sub -- is:

Minimum $\lceil (N-1)/2 \rceil + 1 = X$
Maximum $X+1 = X_1$

Where N = the number of Chip-8 instructions in the sub (including the 00EE return) and X and X_1 = the minimum and maximum number of Chip-8 instructions that will be executed by the subroutine. Can a reader point out why counting Chip-8 instructions may <u>not</u> be an accurate way of evaluating the speed of a subroutine? A future column will deal with the subject.

PROJECT #1 - Chip-8 programmers -- the second advantage in using the BMMM sub above will be explained next month, but can you find it? (Hint -- what happens in each subroutine when V0=00?)

PROJECT #2 - Machine language programmers -- the sub @ 01A4-01B3 controlling the BMMM instruction in the Chip-8 Interpreter is unnecessarily long by two bytes! Can you discover how to shorten the sub? (Hint -- what's a good way to add a <u>possible</u> carry to the high part of a 16-bit address during a multiple precision add?)

PROJECT #3 - Advanced programmers -- Chip-8 does not offer forward and back relative branching -- in other words a jump up or down by the number of <u>instructions</u> specified in a variable. This would facilitate the writing of relocatable code. By replacing the BMMM instruction with a new sub in the Chip-8 interpreter, create two <u>new</u> instructions that provide relative branching to Chip-8. (Hint -- you will probably want to use an FXNN form for the instruction. Also some relocating will have to be done to two other routines to open enough room.) All answers next month.

THE LUNAR LANDER WITH COLOR AND SOUND EFFECTS

By: Wim van Alphen

One of the nicest games of the VIP Game Manual (VP710) is the Lunar Lander program. If you have a VIP with a color board VP590 and a simple sound board VP595, this modification adds nice color and sound effects to the Lunar Lander program. It also changes some graphics a little bit, to make it more attractive. If you only have a color board, you can run this program without problems. You will only miss the sound effects if you land safely or run out of fuel. A VIP without a color board (with or without simple sound board) will NOT play this program, but bombs out.

First you load the Lunar Lander into your VIP (from tape, or type it in from the VIP Game Manual). Then change the memory contents at the memory locations indicated below. Finally, type in the new program, from location Ø9ØØ up to ØA87. If you want to store your Color Lunar Lander on tape, don't forget it is now ØB pages. Try it out and have fun!

Changes:

0200	1928	069E	198A
05EA	19D2	06D6	1A24
05EE	19D2	06EE	2A18
05F2	19D2	072E	2A18
05F6	19D2	0744	1A2A
05FA	19D2	074A	1A3E
05FE	19D2	075C	198A
0616	2A5A	079E	197A
061A	29B6	07EA	194A
0626	29D6	0824	19F2
063E	2A5A	082C	00EE
0642	29B6	085E	1A46
064E	29D6	0874	1A52
0666	1A38	0886	1A2E

Additions:

0900	45BF	093C	0900	0980	197C
02	45AF	3E	D340	82	17A0
04	45AE	40	3000	84	6D40
06	055F	42	0900	86	FD18
08	1F	44	D378	88	1680
09	2E	46	0701	8A	090F
0A	8E3A06	48	1202	8C	D301
0D	15	4A	00E0	8E	0501
0E	D4	4C	0900	90	090F
0F	45BF	4E	D300	92	D302
11	45AF	50	0005	94	0501
13	45AE	52	090F	96	090F
15	055F	54	D307	98	D303
17	8F	56	2006	9A	0501
18	FC08	58	090F	9C	090F
1A	AF	5A	D306	9E	D304
1B	2E	5C	2006	A0	0501
1C	8E3A15	5E	090F	A2	252E
1F	15	60	D366	A4	6F20
20	D4	62	1404	A6	FF15
21	EF65	64	090F	A8	FF07
23	EF65	66	D367	AA	3F00
25	EF65	68	1404	AC	19A8
27	D4	6A	090F	AE	610C
28	0923	6C	D3D6	B0	6200
2A	0900	6E	0601	B2	A47E
2C	D300	70	090F	B4	19A2
2E	0005	72	D3D7	B6	78FF
30	0900	74	0601	B8	6F08
32	D300	76	0921	BA	FF15
34	3007	78	1226	BC	FF07
36	0900	7A	2568	BE	3F00
38	D390	7C	F307	C0	19BC
3A	3006	7E	3300	C2	00EE

Additions (Continued):

09C4	A5C8
C6	6B20
C8	FB15
CA	FB07
CC	3B00
CE	19CA
D0	00EE
D2	7703
D4	163A
D6	7801
D8	090F
DA	D369
DC	0706
DE	090F
E0	D36A
E2	0706
E4	090F
E6	D36B
E8	0706
EA	090F
EC	D36C
EE	0706
F0	00EE
F2	6F20
F4	FF15
F6	FF07
F8	3F00
FA	19F6
FC	A4A2
FE	090F
0A00	D331
02	0504
04	090F
06	D332
0A08	0504
0A	090F
0C	D333
0E	0504
10	090F
12	D334
14	0504
16	1826
18	6F04
1A	FF15
1C	FF07
1E	3F00
20	1A1C
22	00EE
24	6E02
26	0923
28	16D8
2A	0923
2C	16FA
2E	26A8
30	0923
32	1888
34	E563
36	60D4
38	0A34
3A	E000
3C	1984
3E	0A34
40	D000
42	F818
44	174C
46	A677
48	60FF
4A	1860
0A4C	F8F0A6
4F	E663D4
52	0A4C
54	70FF
56	6E02
58	1876
5A	A5C8
5C	603E
5E	0A4C
60	6FFF
62	FF18
64	29C6
66	6037
68	0A4C
6A	FF18
6C	29C6
6E	6045
70	0A4C
72	FF18
74	29C6
76	608B
78	0A4C
7A	FF18
7C	29C6
7E	605D
80	0A4C
82	6F40
84	FF18
86	00EE

A CHEAP PRINTER FOR THE VIP

by Ruben de la Pena

Some months ago, I saw an ad in a computer magazine, offering an alphanumerics thermal printer with 12 Characters; I ordered it and they sent me the Texas Instrument EPN9112 Printer with a TI bulletin; the application basics of the printer. It's actual price is $59.95.

THE PRINTER

It consists of a printhead of 12 groups of 5 dots and a stepping motor, so you can print one line, advance the paper by the stepping motor and so on, until you have formed the character in any desired matrix. I made it 4 x 5, to keep the same format of the VIP, but with minor changes to the software, you can make it in standard ASCII 5x7 characters.

To print a dot, you should use 18 Volts to the selected dot, (1 of 5), and send to ground the common of the group or digit, for 8 Milliseconds. The dots are multiplexed and all you need is to scan the 5 dots with 18 Volts, sending to ground the appropriate digit to write anything you want.

THE INTERFACE

The electronics interface has 4 parts:
1. The dot drivers scanning.
2. The shift register and character drivers.
3. The motor drive circut and
4. The power supply.

As you can see from the diagram, the scanning is controlled by one 4017 I.C. (CMOS Decade counter), and Darlington drivers to take the 18 Volts to each group of dots. From the same 18 V. line, we use a 1K Ohms resistor and a Zener diode to drop the voltage to 4.7 Volts for the IC's. This counter is controlled for the VIP -- with the bits 3 and 4 of the output port.

The shift register and character drivers are formed for two 4015 ICs connected in series to integrate a 12 bit character and coupled with 3 4066 transmission gates to 12 driver transistors. The clock input of the registers goes to bit 2 of our output port, the data input goes to Q and the enable of the gates goes to bit 1. The motor circut consists of drivers to the winding of the stepping motor coupled capacitively. The phase 1 and phase 2 goes to bits 8 and 7 of our port.

The last part is the power supply. I use 2 18 V. 1 Amp. transformers, but if you can get a 36 V.C.T. Transformer, you can use it as well. You will need 18 V. for the stepping motor and 18 Volts for "Burning" the dot in the thermal paper.

You can get Thermal paper in any Radio Shack Store. (Cat. No. 65-706 3 Rolls for $2.79)

THE SOFTWARE

The program is relocatable in any place of memory, in RAM or ROM. It uses page 0E like scratchpad, and has 2 pages of instructions. In the first program, (4x5 format), you need to put the starting address at:
0E90 High byte of Starting address
0E91 Low byte of starting address

Put a long branch to the page of the program (0C) and flip to RUN. The printer will start printing with this format:

XXX XXXXXXXX

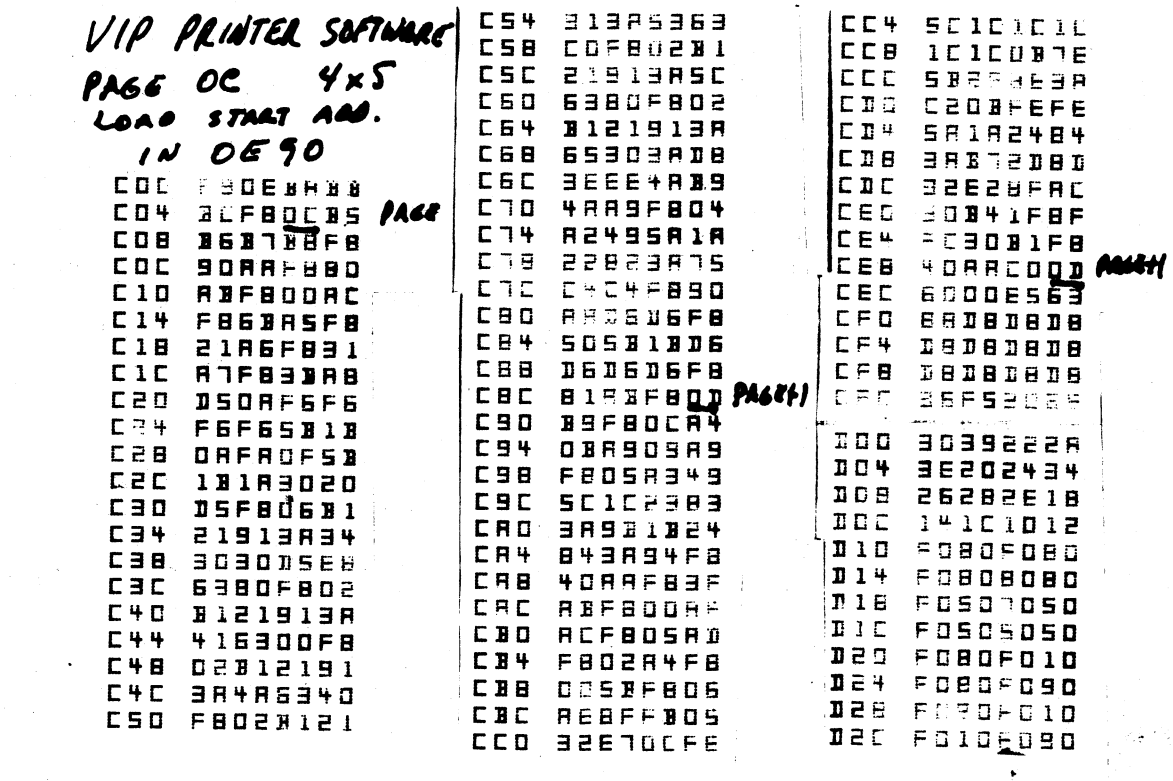
Address Four bytes

The program makes an extra space in each page boundary. To stop, reset your VIP.

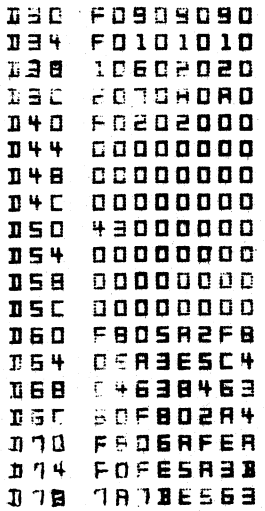
If you press "C" immediately after operating the RUN switch, the printer will advance the paper only. When you release the "C" key, it will start printing at selected address.

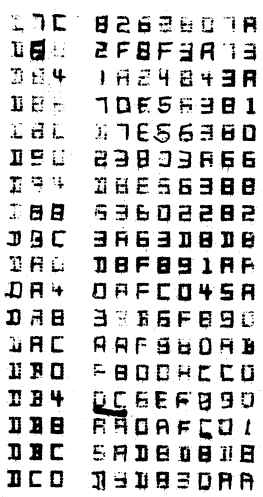
This project needs 3 P.C. boards; 1 for the main circut, 1 for adapting the flexcircut of the printer assembly, and one to plug in the output port.

I can supply the 3 PC (epoxy drilled) for $19.95, PC boards and parts (less printer and power supply) $49.95, and PC boards and parts and printer (less power supply) $109.95 (with assembly instructions). You can also get just the printer from American Micro Products, 6550 Tarnef M/S 11, Houston, TX, 77074 for $59.95. Please send your check or money order to Disenos Electronics, Edison 1210 N., Monterrey, N.L., MEXICO.



2.04.16





PAGE

```
HERE IS A
SAMPLE OF
THE TEXT
PRINTER:
0 1 2 3 4 5
6 7 8 9 A B
C D E F G H
I J K L M N
O P Q R S
T U V W X Y
Z . + - ? *
, = : ! ; (
) $

********
```

VIP PRINTER SOFTWARE

PAGE 0C 5x7

Load Start address IN-0EF0

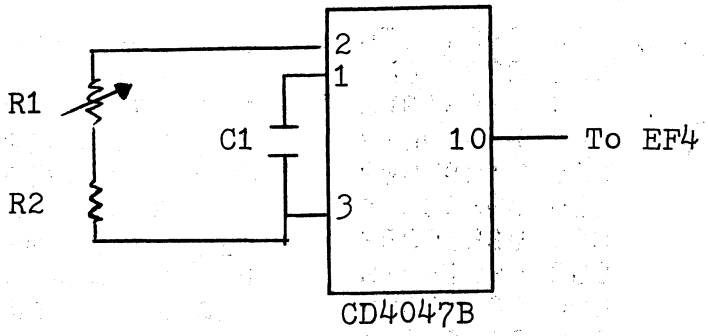
```
C00 F80EBABB
C04 BCF80CB5
C08 B6B7B8F8
C0C F0AAF8E0
C10 ABF800AC
C14 F86BA5F8
C18 21A6F831
C1C A7F83BA8
C20 D50AF6F6
C24 F6F65B1B
C28 0AFA0F5B
C2C 1B1A3020
C30 D5F806B1
C34 21913A34
C38 3030D5E8
C3C 6380F802
C40 B121913A
C44 416300F8
C48 02B12191
C4C 3A4A6340
C50 F802B121
C54 913A5363
C58 C0F802B1
C5C 21913A5C
C60 6380F802
C64 B121913A
C68 65303AD8
C6C 3EEE4AB9
C70 4AA9F804
C74 A2495A1A
C78 22823A75
C7C C4C4F8F0
C80 AAD6D6F8
C84 785B1BD6
C88 D6D6D6F8
C8C E1ABF80D
C90 B9F80CA4
C94 0BA909A9
C98 F807A349
C9C 5C1C2383
CA0 3A9B1B24
CA4 843A94F8
CA8 60AAF85F
CAC ABF800AF
CB0 ACF805AD
CB4 F802A4F8
CB8 005BF806
CBC AE8FFB07
CC0 32E70CFE
CC4 5C1C1C1C
CC8 30F80B7E
CCC 5B2E8E3A
CD0 C20BFEFE
CD4 5A1A2484
CD8 3AB72D8D
CDC 32E28FAC
CE0 30B41F8F
CE4 AC30B1F8
CE8 80AAC00D
CEC 8000E5A3
CF0 88D8D8D8
CF4 36F2306E
CF8 1C1C1C1C
CFC 30CA306E

D00 1D5B2F29
D04 6F3B2355
D08 42464E10
D0C 61163568
D10 F0484870
D14 4848F048
D18 48484848
D1C F0708898
D20 A8C88870
D24 8880F088
D28 88708808
D2C 30088870
D30 88087080
D34 80F88080
D38 E08080F8
D3C 80807008
D40 08F07088
D44 89708888
D48 70888878
D4C 08887088
D50 88F88888
D54 88F88810
D58 20202020
D5C 60202020
D60 20708880
D64 80808870
D68 F88080E0
D6C 8A808010
D70 3050F010
D74 10100000
D78 79000000
D7C 00000000
D80 F860AAF8
D84 07A2F805
D88 A3E56384
D8C 6380F802
D90 A4F806AF
D94 EAF0FE5A
D98 3B9B7BE5
D9C 63826380
DA0 7A2F8F3A
DA4 941A2484
DA8 3A91E563
DAC 81D76380
DB0 23833A89
DB4 D8E56388
DB8 63802282
DBC 3A861D8D8
DC0 D8F8F1AA
DC4 0AFC045A
DC8 32D6F8F0
DCC AAF8E0AB
DD0 F800ACC0
DD4 0C6EF8F0
DD8 AA0AFC01
DDC 5AD8D8D8
DE0 D8D830CA
DE4 00000000
DE8 00000000
DEC 00000000
DF0 00000000
DF4 00000000
DF8 00000000
```

2.04.17

INTEREST INVENTORY/INQUIRY

We can get inexpensive light pens which can be adapted for use on the VIP. The Lipson Light Pen retails normally for $24.95 - but it is intended for use on an APPLE, not on a VIP. So the manual and cassette tape would be next to useless for us.

However - Rick drew up a quick Interface diagram so we can get an idea about how to put the light pen to work on a VIP:



Connect pins 4, 5, 6, and 14 to +5v
Connect pins 7, 8, 9, and 12 to ground

R1 is the light pen - available from ARESCO if enough interest is shown.

R2 =10K

C1=.001 μf or other, depending on use.

Output frequency of this free-funning multivibrator will vary with the amount of light falling on the light pen. For R2=10K and C1=.001 μf, the frequency will range between 2.3 kHz (full dark) to 23kHz (full bright)

If we buy in hundred-lots, we can get the pen for $14.95. At that price, it will include the APPLE manual (it has info in it you might find useful as you work up your own programs), but no programs. Anybody out there want one at that price?

Also - regarding the Trendcom printer mentioned elsewhere in this issue - we just found out that there are 16 rolls of paper in the $48 case - so we can sell the paper for $3.00 per roll.

And, as another aside, I've recently heard that there's an 1802 interface in the works for the Exatron Stringy Floppy. (This is a cassette-looking rapid access storage device). W're waiting for info.....

TONE GENERATOR

by Albert Glasser

Here is a program (machine language) for the 1853 tone generator when it is connected as per the Studio II which I find more convienient.

This program allows the use of the keyboard to hear the tones being played. The tones desired are set up and placed in memory initially. For this program I have used Key C as note C third octave and arrayed the rest above and blow this for almost 2 octaves. They are identified on the program listings. The program also stores in sequence the tone you play as well as how long you play it (up to 24 seconds). Afterwards you can set the machine to replay what you played. One page of memory is needed to store the program, and one page each (to as many desired) for the tones and time.

A rest key is provided and must be hit after you release the key you have just played. It is possible to enter the tone values manually in the tone page without playing them if you put the time in the time page. However, this would require a determination of the Hex value for the time.

The program as supplied is explained as follows:

1. The main program is in ØØ.
2. The tones page is 1F.
3. The time page is 1E.
4. The rest key is Key 2.
5. The "end of program" key is Key 5 and must be used at the end to allow the repeat of the program automatically as well as to let the machine know where you want it to stop playing tones.
6. To execute the program start at ØØØØ.
7. To replay on a VIP the first 3 values must be changed to C0 01 B1 or if a different system allows execute at ØØB1. I'm sure your readers can modify the program to perhaps allow for a key to be used to provide replay. I'm also sure this program is readily altered for other notes, additional ones etc.

I also have a VIP language (CHIP-8) Star Trek game that doesn't have to much action but a lot of graphics. It takes approximately 6 pages to run.

I am using an ELF system modified to an ELF II, VIP and etc. My CHIP-8 is in ROM, my operating system is a modified VIP to allow much more expansion. This system is all wire wrapped and looks it but performs well. Additional items are a 2708 programmer and its program in ROM. I am willing to help others. My kids love the games and my one has wire wrapped a VIP from the schematic... this is why I like to have contact with the VIPER.

One suggestion or critisism is that I went to the VIPER because it appeared it was going to be an exchange of ideas, program, and etc. between people. However, I noticed that

you are now withholding programs for sale. If you want to sell, please still provide the program in the VIPER for those of use who don't mind entering the code. Example, Baseball

```
0000 F8 C0 2nd value for replay or a VIP
  01 1F 00 Page # where tones stored
  02 BC B1 Load 4th Reg. C hi and lo
  03 F8    Change to
  04 00     any
  05 AC     desired
  06 AE    Time reg.
  07 F8
  08 1E    Page for time (note) length
  09 BE
  0A 7A    Q off (tone off)
  0B 69    Video on of desired to simulate
  0C F8    Simultaneous operation
  0D 00
  0E BA    Initialize stack pointer for note output
  0F F8
0010 FF
  11 AA
  12 E2    Set X = 2 = stack keyboard scan routine
  13 F8
  14 0F    Load an 0F into Reg. D0
  15 AD
  16 2D    Decrement Reg. D
  17 22    Dec. stack
  18 8D    Get reg. D low
  19 52    Load into stack
  1A 62    Output to keyboard
  1B E2    Spare
  1C E2    Spare
  1D 3E    B43 is this key pushed if not go get another; if
           yes, debounce
  1E 16
  1F F8
0020 04
  21 A8
  22 28    Dec 8
  23 88    Get reg 8 lo
  24 3A    If not Ø go back
  25 22
  26 8D    If Ø get reg D low which is key that was pushed
  27 FE
  28 FE    Shift left 4X
  29 FE
  2A FE
  2B F6    Shift right 4X
  2C F6    Key that was pushed now = 0X
  2D F6
  2E F6
  2F B8    Store it via reg 8 hi
0030 E2    Waste time
```

```
   31 E2    Spare waste time
   32 EA    Set X = reg A
   33 98    Get key pushed before X or immediate was it an A
   34 FB
   35 0A
   36 32    B7 on acc. = Ø
   37 79    It was an A go get note
   38 98    Opps not an A
   39 FB    May be a "B"
   3A 0B
   3B 32
   3C 7D
   3D 98
   3E FB
   3F 0C    How about a "C"
   40 32
   41 81
   42 98
   43 FB
   44 0D    Or a "D"
   45 32
   46 85
   47 98
   48 FB
   49 0E    Or "E"
   4A 32
   4B 89
   4C 98
   4D FB
   4E 0F    Or a "F"
   4F 32
0050 80
   51 98
   52 FB
   53 00    Or a Ø
   54 32
   55 91
   56 98
   57 FB
   58 01    Or "1"
   59 32
   5A 95
   5B 98
   5C FB
   5D 09    Or "9"
   5E 32
   5F 99
0060 98
   61 FB
   62 08    Or "8"
   63 32
   64 90
```

```
0065 98
  66 FB
  67 04     Or perhaps a "4"
  68 32
  69 A1
  6A 98
  6B FB
  6C 07     Or a "7"
  6D 32
  6E A5
  6F 98
  70 FB
  71 05     Or "5" 5 is used for end of program
  72 32
  73 A9
  74 98
  75 FB
  76 02     Or "2" will be rest note
  77 32
  78 AD
  79 F8
  7A 7C     Note = an A3  3rd octave
  7B 30     Any numbers can now be put in these places
  7C D0
  7D F8
  7E 6F     =B3
  7F 30     B♯ to output routine
0080 D0
  81 F8
  82 68     = C4
  83 30
  84 D0
  85 F8
  86 5D     = D4
  87 30
  88 D0
  89 F8
  8A 53     =E4
  8B 30
  8C D0
  8D F8
  8E 4E     =F4
  8F 30
0090 D0
  91 F8
  92 45     =C4
  93 30
  94 D0
  95 F8
  96 8A     =G3
  97 30
  98 D0
```

```
  99 F8
  9A 3E    =A4
  9B 30
  9C D0
  9D F8
  9E 37    =B4
  9F 30
00AØ D0
  A1 F8
  A2 90    =F3
  A3 30
  A4 D0
  A5 F8
  A6 34    =C5
  A7 30
  A8 D0
  A9 F8
  AA ØØ    End of program mark
  AB 30
  AC D0
  AD F8
  AE 01    Rest note
  AF 30
00B0 D0
  B1 F8    Initialize page where note length stored
  B2 1E
  B3 B8    Can be any desired page
  B4 F8
  B5 00
  B6 A8
  B7 AA    Intialize where tones are stored
  B8 AC
  B9 F8
  BA 1F
  BB BC
  BC EC    Set X
  BD 69    Turn on video if it was on for storage of notes
           and time and Q off
  BE 7A
  BF 48
00C0 BA    Load and advance thime put in reg A high
  C1 64    Output what's at X = reg C
  C2 7B    Turn on Q and tone
  C3 2A    Dec where time is stored and get time
  C4 9A
  C5 3A    If not a 0 end of program go back and dec time
  C6 C3
  C7 2C    Dec reg C tones to set original tone just pre-
           viously outputted and put in accumulator then
           increment C to point to next note if note last
           and in acc. = Ø.
  C8 FØ
  C9 1C
  CA 32
  CB F6
  CC 30
```

```
  CD BE
  CE 00
  CF 00
00D0 5C     Come here to putput note to tone generator load
            tone into memory pointed to by rge C and reg A
  D1 5A
  D2 1C
  D3 F8
  D4 00
  D5 BD
  D6 AD
  D7 3E     If key not down BN3 to restart X = A set previous-
            ly
  D8 0A
  D9 64     Output note via X = A
  DA 7B     Turn on Q and tone
  DB 2A     Dec A = stack back to __FF
  DC 1D     Inc D = time since key still down and tone on
  DD 36     Is key down BZ 3
  DE DC
  DF 9D     Get time store via reg E points to 1E00 first
            then is incremented
00E0 5E
  E1 1E
  E2 30     Branch to beginning to get next note
  E3 0A
  E4 00
  E5 00
  E6 00
  E7 00
  E8 00
  E9 00
  EA 00
  EB 00
  EC 00
  ED 00
  EE 00
  EF 00
  F0 00
  F1 00
  F2 00
  F3 00
  F4 00
  F5 00
  F6 F8     Time out before repeat of tones stored
  F7 A0
  F8 B8
  F9 28
  FA 98
  FB 3A
  FC F9
  FD 30
  FE B1
  FF STACK
```

ELFs CAN READ VIP TAPES

P V Piescik

(Courtesy of Cuddly Software, New York)

A small modification to the VIP ROM ('revealed' in the VIPER Sept '78 issue (Vol. I, #3) pages 13-23) will compensate for the higher clock frequency on both the Netronics' ELF II and the Quest Super ELF, to allow these machines to read tapes created by RCA VIPs. This patch was created upon an inquiry from a customer, who has since reported that it does work as intended.

The problem lies within the timing of the tape signal; the timing loop is executed faster on ELFs due to the 1.65% faster clock frequency. By my calculations, the VIP timing assumes that a signal which lasts 75% of the longer theoretical duration (lower frequency) must be a '1' bit; and anything shorter must be a '0' bit (shorter duration, higher frequency). Since this 75% is 250% of the shorter duration, it would appear that the timing is loose enough to accomodate the variation in the clock. Empirically, this is disproven.

Since the timing is done in a 6-cycle loop, changing the number of iterations by ± 1 results in a resolution of ± 6 cycles by which the loop duration may be varied. Increasing the number of iterations to compensate for the faster clock, we'd like to increase the count from its original 16 (dec) to 16.264; but, dealing with integers, we're forced to round this down to 16, which is where we started (and got nowhere).

Second attack: the original loop totals 96 cycles, from initiation of the iteration count to the exit from signal detection; if we stretch the loop to compensate for the fast clock, we need a total of 97.584 cycles, which rounds up to 98! Now, if we add a single 2-cycle instruction at a point where it's executed only once per bit, we should be home free. So, the solution is to add a SEX 2 (E2) at 8187, and shove down the rest of the code which follows. It so happens that RCA cut us some slack (4 bytes of '00' at 81B5-81B8), so we do have the room. So, plugging in the following changes will allow ELFs to read VIP tapes!

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
8180			D3	F8	10	3D	85	E2	3D	90	FF	01	3A	88	17	9C
8190	FE	35	91	30	82	D3	E2	9C	AF	2F	22	8F	52	62	E2	E2
81A0	3E	99	F8	04	A8	88	3A	A5	F8	04	A8	36	A8	88	31	AB
81B0	8F	FA	0F	52	30	95										

The idealists among you will also want to make the following change to correctly address the keypad debounce routine, although you can get by without it:

806D A2

A couple of other thoughts come to mind....if you try to read more data than is on the tape, or the VIP routines detect a parity error, the Q LED will light upon detection of the condition. However: reading does continue! I haven't determined what happens to the memory pointer (R6) for short tapes, but I do know that subsequent tape files (if any) will be appended until the desired number of pages (which you keyed in) has been read. I can only recommend caution.

A PROUD ANNOUNCEMENT FROM ARESCO

On September 4, 1979, ARESCO opened a computer store in the Columbia, MD area. We're located at 9143 G Red Branch Road, Columbia, MD 21045. Any VIP people in the area are invited to drop by and chat...although the store doesn't stock VIP products (the Computer Center of Columbia is an APPLE Center), we have our newsletter activities going on in the back room. Volunteers for hard labor (such as entering programs so we can test them out before printing them) or harder labor (such as proofreading the code once it's been entered) or hardest labor (such as helping to address all those envelopes) are more than invited: you are actively encouraged to lend as many hands as you possess!

For those of you who don't know, ARESCO publishes four other newsletters each month - so we're kept pretty busy. It would be nice to be able to say we're getting rich and can even pay for real hired hands....but you wouldn't want me to lie to you, would you? We have a few people helping out; mostly folks who are interested in one aspect of computing or another, but not necessarily in the VIP. HELP anyone???

And yet another announcement: the Studio II RAM card is off to production! We expect to get it back around the 21st of October, and will sell it without the 2114 RAM chips for $20. If you have a Studio II video game, and haven't gotten around to climbing on the Add-a-PROM bandwagon, better write and let us know. The conversion information package (and the introductory issue of the STUDIO TUTOR newsletter) is $5, and we have the PROM ($15), the PCB ($10), and a backplane board with ($36) or without ($20) connectors. A subscription to the newsletter is $10 for 6 issues - and issue #3 is being mailed out in just a few days. That newsletter was a direct result of the large number of you who wanted SII news kept separate from the VIPER (although you didn't object in as much volume to occasional ELF conversion articles).

You'll need the 1802 manual and the 1861 data sheet if you get serious about the conversion...we can supply you with all that stuff...just send in your order & check and we'll get it all out to you within two weeks (except the RAM card; it will be awhile on that one yet.)

GET READY FOR MORE PIPS!

Due to the enthusiastic response to PIPS I and PIPs II, Tom Swan has prepared Volume III of PIPs. Just as in the first two volumes, you'll find the same detailed documentation, the same carefully commented source code, and the same thoughtful explanations of how you can modify the programs to meet your own desires. And all of the information is delivered in the unique and readable style that is the hallmark of all of Tom's work.

Volume III is devoted to just two programs - two of the best games we have ever seen on the VIP: VIP-FLOP and VIP-OKER. VIP-OKER lets you sit in on a game of five-card draw poker with three computerized opponents (Rick, Terry, and Tom!). Each player has a different strategy; one bluffs a lot, one bluffs hardly ever, and one bluffs only now and than. Naturally, you don't know which player has which style (it changes each time you re-run the program), but each will apply his (or her) strategy consistently throughout the game. The program follows all the rules of draw poker, and you can examine the program to be sure the computer isn't peeking at your hand!

As for VIP-FLOP, here's what Tom himself has to say about it in the introduction to the book:

"Othello,* the box game similar to your COSMAC VIP-FLOP, has been a popular game on computers for several reasons. For one, it isn't terribly difficult to program, although the method of figuring the computer's move duplicates chess, checkers, and other games, using a look-ahead feature to make up the computer's mind. Also, the complexity of manipulating the pieces during play - there's a good chance that players will make errors in flopping poker chips on a board - make this game an excellent choice for a television display. The computer handles all these chores, so the board is always "right" - something a box game cannot do.

"VIP-FLOP is supplied in three versions. You can play against the computer, you can play with a friend, or you can sit and watch while the VIP plays against itself, a feature that can be used to demonstrate the game to a beginner or as a way to test new evaluation routines you might write yourself and in sert. Nine levels of play are possible and the VIP will require from one (or two) seconds to as long as 15 minutes to figure its next move. At the highest level, the program looks eight moves ahead - and is a very tough customer to beat. When you play with your VIP, you can let the computer go first; if you're stumped during any part of the game, you can even ask the computer to recommend a move for you!"

---

*Othello is a Registered Trademark of CBS Corporation

So there you have it....two more of Tom's finest. As with the two previous volumes, we offer a pre-publication special (to raise the cash to go to press with the book). You can get the third volume of PIPS with a cassette tape for $14.95 until January 15, 1980. After January 15, the price will go up to $19.95 with the cassette.

Please ship me the following Tom Swan PIPS FOR VIPS volumes. I enclose payment in full for my entire order (or charge my Visa/BankAmericard, Master Charge, or American Express card).

PIPS I with cassette ($19.95)_____/without cassette (14.95)_____
PIPS II with cassette ($19.95)_____/without cassette (14.95)_____
PIPS III with cassette ($14.95)_____/

Please note that you can purchase the books without the tapes, but you can't purchase the tapes without the books.

NAME (Please type or print)_____

ADDRESS (street - for UPS delivery)_____

CITY_____STATE_____ZIP_____

Please bill my credit card (NAME OF CARD)_____

CARD HOLDER'S NAME (if different)_____

CARD NUMBER_____EXP DATE_____

MC INTERBANK # (the 4 digits above your name)_____

REQUIRED CREDIT CARD SIGNATURE_____

NEW PRODUCT ANNOUNCEMENT

Operating Systems:
Concepts and Principles
Price: $6.95 (152 pages)

Microcomputer Applications
P.O. Box E
Suisun City, CA 94585

Microcomputer Applications announces a new publication: Operating Systems: Concepts and Principles, the first volume of the Microprecessor Software Engineering Concepts Series focuses on the topic of perating system software. This book presents an overview of the basic operating system types, their components, and their capabilities. In addition, resource management, file systems, and task communication are explained in depth. The detailed glossary contains over 200 terms (task, process, mailbox, swapping, etc.).
Contents: Operating System Introduction and Overview; Tasks and Contexts; System Services and Support; Communication and Synchronization; Scheduling; Resource and Memory Management; I/O and File Systems; System Security

For order and delivery information write to Microcomputer Applications, Dept. S-1, P.O. Box E, Suisun City, CA, 94585. CA residents add 6% sales tax. Please add $.75 per book for postage and handling.

ORDER FORM * ORDER FORM * ORDER FORM * ORDER FORM * ORDER FORM

TO ORDER RCA PRODUCTS: Fill out the order form at the bottom of the page, indicating which RCA item you wish to order. Then handwrite a note to the effect that you are aware that delivery of RCA produced items may take more than 30 days and that we are authorized to hold your money until RCA can deliver to us. We will hold it - we won't cash your check or charge your credit card until the day we make shipment to you.

TO ORDER ARESCO PRODUCTS: Just fill out the form. All ARESCO products are available within two days of our receipt of your order, and we can ship immediately.

Please ship me the items listed below. I enclose full payment or authorize you to charge my MC/VISA credit card.

NAME_____ MC/VISA #_____

STREET_____ EXP. DATE _____

CITY, STATE, ZIP_____MC INTERBANK #_____

CREDIT CARD ORDERS REQUIRE SIGNATURE_____

_____

_____

_____

_____

_____

_____

_____

_____

Mail order to:

ARESCO
Box 1142
Columbia
MD 21044

2.04.29